Journal of Applied and Computational Sciences in Mechanics, 37, 1, 2025. (53-70)



Ferdowsi University of Mashhad

Journal of Applied and Computational Sciences in Mechanics

Page Journal: mechanic-ferdowsi.um.ac.ir



Society of Manufacturing Engineering of Iran

Optimization of Engineering Problems with Discrete and Continuous Constraints Using Dynamic Adaptive Meta-Heuristic Algorithms*

Research Article

SayedMostafa Mousavi[1], Alireza Toloei [2], MohammadHosein Toloei[3]

10.22067/jacsm.2024.86041.1232

Abstract

This article examines and implements meta-heuristic algorithms to solve five engineering problems. The problems have been investigated in a constrained manner and optimized using firefly, colonial competition, frog, ant and gray wolf algorithms. For each algorithm, a dynamic adaptive factor is introduced to balance the convergence rate and optimal search. The results show that each algorithm performs better in specific problems: firefly in spring tension, frog in three-bar truss, colonial competition in speed reducer and gear design, and gray wolf in pressure tank. The article concludes that choosing the right algorithm depends on the type of problem and environmental conditions.

Key words: Constrained engineering design optimization problems, continuous and discrete design variables, meta-heuristics, Imperialist Competitive, ants, Frog, grey wolf , firefly.

1. Introduction

Numerous optimization algorithms have been proposed to solve various engineering design optimization problems, which usually involve nonlinear constraints. Optimization algorithms can generally be divided into two categories: stochastic algorithms and deterministic algorithms. Traditional optimization methods, such as the gradient descent method, quasi-Newton methods, and the interior-point Newton method, are usually based on gradients and require different algorithms and conditions depending on the objective functions. However, these methods are often inefficient and inaccurate when dealing with complex, high-dimensional, and nonlinear optimization problems, especially when objective functions and constraints are discontinuous. To overcome these issues, many stochastic optimization algorithms have been proposed, including particle swarm optimization, genetic algorithms, firefly algorithms, ant colony optimization, artificial bee colony optimization, frog leaping algorithms, imperialist competitive algorithms, and grey wolf optimization algorithms. These stochastic optimization methods are typically meta-heuristics inspired by physical and natural phenomena.

2. Solution Methods

Meta-heuristic algorithms are a set of advanced methods used in optimization and complex problem-solving. These algorithms are inspired by natural and social processes and behaviors, such as food searching in animal societies. The main goal of meta-heuristic algorithms is to find the most optimal solutions within the search space and to avoid getting trapped in local optima. Some of these methods, such as the ant colony optimization algorithm, are particularly suitable for problems with discrete variables. The five solution methods selected in this paper are suitable for problems with continuous variables, discrete variables, or a combination of both.

2.1 Dynamic Adaptive Factor

The dynamic adaptive factor is introduced to balance the convergence rate and the global search ability during the optimization process. This factor adjusts the search speed according to the problem environment, aiming for better optimization by dynamically adapting to environmental conditions. Through this dynamic adjustment, the algorithms are able to enhance the convergence rate and perform a more effective search within the solution space.

3. Introduction to Engineering Optimization Problems

3.1 Introduction to Constrained Engineering Optimization Problems

In this section, five constrained engineering optimization problems—each involving objective functions, design variables, and various constraints—are selected. The performance of the firefly algorithm, the imperialist competitive algorithm, and other proposed algorithms is

*Manuscript received December 24, 2023. Revised July 9, 2024, Accepted September 11, 2024.

[1] PhD student in Aerospace Engineering, Shahid Beheshti University of Tehran, Tehran.

[2] Corresponding author, Associate Professor, Shahid Beheshti University of Tehran, Tehran. **Email:** toloei@sbu.ac.ir

[3] PhD student in Project and Construction Management, Tehran University of Arts, Tehran.

evaluated in terms of solution quality, stability, and convergence rate.

3.2 Constrained Optimization Problems with Continuous Variables

- Tension/Compression Spring Design Problem: The objective of the spring design problem is to minimize its weight while satisfying constraints on minimum deflection, shear stress, frequency, and maximum outer diameter.

- Three-Bar Symmetrical Truss Design Problem: The three-bar symmetrical truss structure, made of steel, is subjected to two constant loads P1=P2=PP_1 = P_2 = PP1 =P2=P. The goal is to achieve the optimal design by minimizing the volume subject to stress constraints. The design variables are the cross-sectional areas of the three bars, denoted as x1x_1x1, x2x_2x2, and x3x_3x3.

- Speed Reducer Design Problem: The objective is to minimize the weight of a speed reducer while considering strength constraints related to the bending stress and surface stress of the gear teeth, as well as shaft stresses and deflections.

- Pressure Tank Design Problem: The aim is to minimize the total construction cost, including material, forming, and welding costs. The design variables are the shell thickness (TsT_sTs), head thickness (ThT_hTh), inner radius (RRR), and cylindrical section length (LLL) of the tank. TsT_sTs and ThT_hTh are discrete variables and must be integer multiples of 0.0625.

3.3 Engineering Optimization Problems with Discrete Variables - Gear Set Design Problem: The goal is to minimize the error between the obtained gear ratio and the required gear ratio (1/6.931), under the constraint that the number of teeth of the four gears must fall within specified permissible ranges.

4. Results

4.1 Sensitivity Test

Generally, the convergence number in a meta-heuristic algorithm indicates the number of iterations required to reach the global optimum. A lower convergence number signifies faster and better convergence to the optimal solution. In this study, sensitivity tests are conducted on the key parameters of the meta-heuristic algorithms to determine the best parameter settings that achieve optimal convergence for all problems.
For instance, in the frog-leaping algorithm used for the three-bar truss design problem, parameters such as the maximum number of iterations, initial population of frogs, number of memeplexes, and size of memeplexes have been evaluated. The final results of the sensitivity analysis for the frog algorithm are shown in Figure 1 alongside other meta-heuristic algorithms.

4.2 Execution Speed Evaluation

According to Table 1, the execution speeds of five meta-heuristic algorithms are evaluated for the three-bar truss problem, based on a fixed fitness value achieved within 100 iterations.

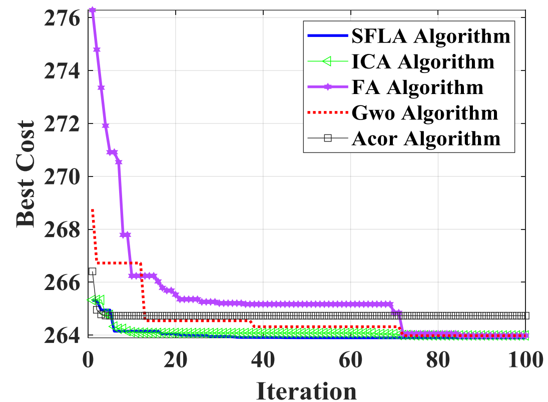


Fig 1. Convergence history of the cost function of the truss design problem

Table 1- The result of the test of the average execution time of the algorithms

	Algorithm type	**Execution time**
1	gray wolf	**18**
2	Colonial competition	**24**
3	firefly	**25**
4	frog	**40**
5	Ant colony	**42**

5. Conclusion

In this article, five problems in the field of mechanical engineering have been examined along with their limitations and related results. Gray wolf algorithms, ant colony algorithm, firefly algorithm, colonial competition and frog algorithms have been evaluated in solving these problems and the relevant results have been described. One of the innovations of this article is satisfying the constraints of the problems, which is done initially due to the limited search range and creating a random neighborhood in the area that satisfies the constraints. Another point is the use of the dynamic adaptive factor in all these algorithms, so that by changing the search speed and adapting it to the environmental conditions, it leads to better optimization.

The results of these algorithms are compared with the results of other researchers in Tables 1 to 5. As an example, the results of solving the three-bar truss design problem by five meta-heuristic optimization algorithms show that the frog algorithm has a more accurate answer in this problem with an accuracy of 0.136%. In fact, for different problems, according to whether the problem is discrete or continuous, a specific optimization algorithm provides a more accurate answer for each problem. These results show that the above algorithms can improve accuracy and speed in different problems at the same time, and this is a manifestation of Swarm intelligence.



علوم کاربردی و محاسباتی در مکانیک

http://mechanic-ferdowsi.um.ac.ir


انجمن مهندسی ساخت و تولید ایران

بهینه‌سازی مسائل مهندسی با قیود گسسته و پیوسته به‌کمک الگوریتم‌های فراابتکاری تطبیقی پویا*

مقاله پژوهشی

سیدمصطفی موسوی(۱) علیرضا طلوعی(۲) محمدحسین طلوعی(۳)

10.22067/jacsm.2024.86041.1232

چکیده *هدف این مقاله، پیاده‌سازی و بررسی روش‌های مختلف الگوریتم‌های فراابتکاری برای حل پنج مسئله مهندسی است. لذا این مسائل مهندسی، به صورت مقید، بررسی شده سپس با استفاده از پنج الگوریتم فراابتکاری کرم شب‌تاب، رقابت استعماری، قورباغه، مورچگان و گرگ خاکستری و با هدف کاهش هزینه‌های مسائل مهندسی، بهینه‌سازی شده و نتایج حل آن صحت‌سنجی شده است. در واقع، در هر یک از الگوریتم‌های بهینه‌سازی استفاده شده، یک عامل تطبیقی پویا برای متعادل‌کردن نرخ همگرایی و توانایی جستجوی بهینه مطلق با تنظیم سرعت جستجو در طول فرایند جستجو معرفی شده است. بررسی‌ها نشان می‌دهد که در هر یک از الگوریتم‌ها، تکنیک‌هایی برای خروج از بهینه محلی استفاده می‌شود که باعث می‌شود پاسخ‌ها به بهینه مطلق همگرا شوند. برای ارزیابی کیفیت و دقت الگوریتم‌ها، از تست حساسیت‌سنجی و مقایسه اعداد همگرایی برای نتایج حاصل از اجرای هر الگوریتم بر روی داده‌ها استفاده شده است. نتایج به دست آمده نشان می‌دهد که الگوریتم کرم شب‌تاب در مسئله فنرکششی، الگوریتم قورباغه در مسئله خرپای سه میله‌ای، الگوریتم رقابت استعماری در مسائل کاهنده سرعت و طراحی چرخدنده، و الگوریتم گرگ خاکستری در مسئله مخزن تحت فشار، در یافتن بهینه مطلق، عملکرد دقیق‌تری را ارائه کردند. در واقع این الگوریتم‌ها برای حل بهینه مسائل با تولید جمعیت تصادفی، ایجاد همسایگی و انتخاب بهترین همسایه به شرط ارضای قیود متغیرهای مسائل، رسیدن به حل بهینه را آسان می‌کند. در نتیجه، این مقاله نشان می‌دهد که هر الگوریتم فراابتکاری می‌تواند در یک مسئله مهندسی مشخص، بسته به نوع مسئله و شرایط محیطی، عملکرد بهتری داشته باشد.*

واژه‌های کلیدی مسائل بهینه‌سازی طراحی مهندسی مقید، متغیرهای طراحی پیوسته و گسسته، فراابتکاری، الگوریتم بهینه‌سازی رقابت استعماری، مورچگان، قورباغه، گرگ خاکستری، کرم شب‌تاب.

مقدمه

تا به حال الگوریتم‌های بهینه‌سازی بی‌شماری جهت حل مسائل متنوع بهینه‌سازی طراحی مهندسی پیشنهاد شده‌اند، مسائلی که معمولا شامل قیود غیرخطی می‌شوند.

الگوریتم‌های بهینه‌سازی را می‌توان تقریبا به دو دسته تقسیم کرد: الگوریتم تصادفی و الگوریتم معین. روش‌های سنتی بهینه‌سازی معین مانند روش شیب نزولی (Steepest Descend)، روش شبه نیوتنی و روش نیوتن بازگشتی داخلی، که معمولا مبتنی بر گرادیان هستند و الگوریتم‌ها و شرایط متفاوت توابع هدف مورد نیاز است.

این روش‌ها برای مسائل بهینه‌سازی پیچیده و غیرخطی با ابعاد بالا، به‌ویژه زمانی‌که توابع و قیود هدف ناپیوسته هستند، ناکارآمد و نادقیق هستند [1].

الگوریتم‌های بهینه‌سازی تصادفی متعددی مانند الگوریتم بهینه‌سازی ازدحام ذرات [2] (PSO : Particle Swarm Optimization)، الگوریتم ژنتیک [3,4]، الگوریتم کرم شب‌تاب [6]، بهینه‌سازی کلونی مورچگان (Ant Colony Optimization Algorithm) [7]، کلونی زنبورهای مصنوعی [8,24,25]، قورباغه

---

* تاریخ دریافت مقاله ۱۴۰۲/۱۰/۳ و تاریخ پذیرش آن ۱۴۰۳/۶/۱۲ می باشد.

(۱) دانشجوی دکتری مهندسی هوافضا دانشگاه شهید بهشتی تهران، تهران.

(۲) نویسندهٔ مسئول، دانشیار دانشگاه شهید بهشتی تهران، تهران.

(۳) دانشجوی دکتری مدیریت پروژه و ساخت دانشگاه هنر تهران، تهران.

Email: Toloei@ihu.ac.ir

[9] (Shuffled Frog Leaping Algorithm)، رقابت استعماری [10] (Imperialist Competitive Algorithm)، الگوریتم بهینه-سازی گرگ خاکستری [5] (Grey Wolf Optimizer) برای غلبه بر این موارد پیشنهاد شده است. این الگوریتم‌های بهینه‌سازی تصادفی معمولا فراابتکاری هستند و از پدیده‌های فیزیکی و طبیعی الهام می‌گیرند.

مطابق مقالات دوریگو (Marco Dorigo) الگوریتم بهینه‌سازی مورچگان الهام گرفته از رفتار مورچگان در جستجوی منابع غذایی است و در سال ۱۹۹۱ توسعه یافته است.

این الگوریتم بهبودیابی، مبتنی بر جمعیت است که به صورت متوالی بر روی یک مسئله بهینه‌سازی عمل می‌کند و قدرت بالایی در بهینه‌سازی مسائل پیچیده دارد. لذا با استفاده از اطلاعات جمع‌آوری شده توسط مورچگان، قادر است به راه‌حل‌های بهینه نزدیک شود و با تکرار این فرایند، بهبودهای اضافی را نیز ایجاد کند [7].

همچنین میرجلیلی در سال ۲۰۱۴، الگوریتم بهینه‌سازی گرگ خاکستری را مطرح کرد که الهام گرفته از رفتار اجتماعی گرگ‌های خاکستری در طبیعت است. این الگوریتم برای حل مسائل بهینه‌سازی و جستجوی فضاهای چندمعیاره استفاده می‌شود و قابلیت همگرایی سریع و جستجوی مؤثر را داراست و می‌تواند بهترین حل ممکن را در فضای جستجوی بزرگ و پیچیده پیدا کند.

با اجرای مراحل تعامل و بهبود مکان گرگ‌ها، الگوریتم قادر است به نتایج بهتر و بهینه‌تر دست یابد و در مسائل پیچیده‌تر نیز عملکرد خوبی داشته باشد [5,11].

در سال ۲۰۰۷ نیز گارگاری و لوکاس (E Atashpaz - C Lucas &Gargari) الگوریتم رقابت استعماری را ارائه کردند که یک الگوریتم فراابتکاری، الهام گرفته از رفتار رقابت و همکاری بین جامعه‌های استعماری در طبیعت است. این الگوریتم برای حل مسائل بهینه‌سازی و جستجوی فضاهای چندمعیاره مورد استفاده قرار می‌گیرد [12,13]. در سال ۲۰۰۳ نیز، مظفر و همکاران، الگوریتم قورباغه را ارائه نمودند که بر پایه رفتار قورباغه‌ها در محیط طبیعی برای حل مسائل بهینه‌سازی مورد استفاده قرار گرفته شده است. پس از آن نیز در بسیاری از مسائل مهندسی مکانیک با هدف کاهش هزینه‌ها استفاده شده است [14-16].

از طرفی در سال ۲۰۰۵، الگوریتم کلونی زنبورهای مصنوعی توسط کارابوگا (Karaboga)، برای بهینه‌سازی پارامترهای مسئله ارائه شد. این الگوریتم به طرز مشابه با رفتار و سازمان‌دهی کلونی‌های زنبورهای عسل در طبیعت برای حل مسائل بهینه‌سازی استفاده شده است [25,26].

در میان همه این الگوریتم‌های بهینه‌سازی تصادفی، الگوریتم بهینه‌سازی کرم شب‌تاب (Firefly Algorithm) به طور گسترده برای حل مسائل بهینه‌سازی مختلف استفاده می‌شود و برای اولین بار در سال ۲۰۰۷ توسط این شی یانگ (Xin-She Yang) معرفی و در محاسبات مهندسی کارآمد نشان داده شده است. از ویژگی-های آن پیاده‌سازی آسان و قابل اعتماد در جستجوی بهینه مطلق است [6,17,18].

در این مقاله از الگوریتم کرم شب‌تاب استفاده شده که یک الگوریتم بهینه‌سازی نشئت گرفته شده از رفتار گروهی کرم‌ها است و برای حل مسائل بهینه‌سازی مورد استفاده قرار می‌گیرد [19,20].

این الگوریتم بر اساس رفتار اجتماعی و تعامل بین کرم‌ها در محیط طبیعی الهام گرفته شده است. از آنجا که مسائل هوافضایی به دلیل پیچیدگی و ابعاد بالا، نیازمند الگوریتم‌های بهینه‌سازی قوی هستند، الگوریتم کرم شب‌تاب به عنوان یکی از ابزارهای قدرتمند در این حوزه استفاده می‌شود. این الگوریتم به‌طور گسترده در مسائل مربوط به علوم هوافضا مورد استفاده قرار گرفته است [21,22].

روش‌های حل

الگوریتم‌های فراابتکاری یک مجموعه از روش‌ها و الگوریتم‌های پیشرفته در حوزه بهینه‌سازی و جستجوی مسائل پیچیده هستند. این الگوریتم‌ها بر اساس الهام از فرایندها و رفتارهای طبیعی و اجتماعی مانند جستجوی غذا در جوامع حیوانی، پدیدار شده‌اند. هدف اصلی الگوریتم‌های فراابتکاری، یافتن بهینه‌ترین راه حل‌ها در فضای جستجو و خروج از نقاط بهینه محلی می‌باشد.

بعضی از این روش‌ها، مثلا روش کلونی مورچگان مناسب برای مسائل با متغیرهای گسسته هستند. پنج روش حلی که برای این مقاله انتخاب شده‌اند، برای مسائل با متغیرهای پیوسته، گسسته و یا ترکیب آن‌ها مناسب هستند.

الگوریتم گرگ خاکستری

این الگوریتم در سال ۲۰۱۴ توسط سید علی میرجلیلی ارائه شده است. الگوریتم گرگ خاکستری یک الگوریتم متاهیورستیک است که از ساختار سلسله مراتبی (Hieratical) و رفتار اجتماعی گرگ‌های خاکستری در هنگام شکار کردن الهام گرفته است. در الگوریتم گرگ خاکستری، جمعیتی از گرگ‌ها به صورت تصادفی در فضای جستجو قرار می‌گیرند. هر گرگ مشخصه‌هایی مانند موقعیت، سرعت و شکار موفقیت‌آمیز خود را دارد. این مشخصه‌ها بر اساس قوانینی که بر اساس رفتار گروهی گرگ‌ها تعیین شده‌اند، تغییر می‌کنند. در هر مرحله از الگوریتم، گرگ‌ها به دو عملکرد اصلی تکیه می‌کنند: اکتشاف و بهره‌برداری. اکتشاف به معنای جستجو در مناطق جدید فضای جستجو و کشف نقاط بهتر است، در حالی که بهره‌برداری به معنای بهره‌برداری از نقاط قوی و بهترین‌هاست [5].

در الگوریتم گرگ خاکستری، گروه گرگ‌ها با توجه به رفتارشان در شکار و اجتماع، به چهار نوع تقسیم می‌شوند. این انواع شامل موارد زیر هستند:

گرگ آلفا (Alpha): گرگ آلفا به عنوان رهبر گروه عمل می‌کند و معمولا بهترین عملکرد را دارد. موقعیت آلفا به عنوان بهترین حل موجود در جمعیت در نظر گرفته می‌شود و بقیه گرگ‌ها بر اساس آن بهبود می‌یابند.

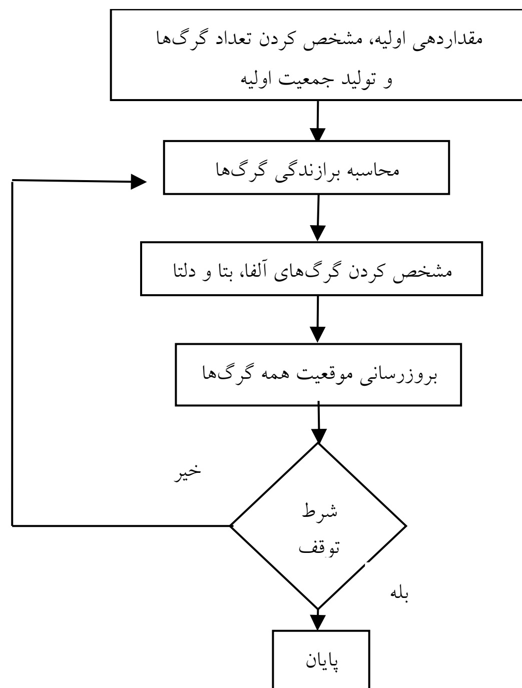
گرگ بتا (Beta): گرگ بتا مسئول کمک به گرگ آلفا در جستجوی بهترین حل است. معمولا در نزدیکی گرگ آلفا قرار می‌گیرد و سعی می‌کند بهترین راهنمایی‌ها را در جهت بهبود عملکرد ارائه دهد.

گرگ دلتا (Delta): گرگ دلتا نقشی مشابه گرگ بتا دارد و به گرگ آلفا و بتا در جستجوی بهترین حل کمک می‌کند. در نزدیکی گرگ بتا قرار می‌گیرد و تلاش می‌کند با تجمیع راهنمایی‌های مختلف بهبودی در عملکرد داشته باشد.

گرگ اپسیلون (Epsilon): گرگ اپسیلون به عنوان گرگی عادی در جمعیت حضور دارد و بر اساس رفتار سایر گرگ‌ها تغییر می‌کند. این گرگ‌ها به عنوان جستجوگران اصلی در جستجوی مناطق جدید و اکتشافی عمل می‌کنند [11].

روندنمای الگوریتم گرگ‌های خاکستری

در شکل (۱) روندنمای کلی الگوریتم گرگ‌های خاکستری مشاهده می‌شود.



شکل ۱ روندنمای الگوریتم بهینه‌سازی گرگ خاکستری [5]

الگوریتم کلونی مورچگان

این الگوریتم یک الگوریتم فراابتکاری بر مبنای رفتار و ارتباطات مورچگان در جستجوی غذا می‌باشد و در دهه ۱۹۹۰ توسط مارک دوریگو و همکاران توسعه داده شد و بر اساس تعاون مورچگان در سفرهای غذاگیری الهام گرفته شده است [7].

در الگوریتم کلونی مورچگان، مورچگان به صورت تصادفی در فضای جستجو قرار می‌گیرند و مواد شیمیایی به نام فرومون‌ها را رها می‌کنند. این فرومون‌ها ارتباطاتی بین مورچگان ایجاد می‌کنند و به عنوان راهنمایی برای انتخاب مسیرها و جستجوی بهینه عمل می‌کنند. در حقیقت الگوریتم کلونی مورچگان از رفتار واقعی مورچه‌ها الهام گرفته شده است، اما شبیه‌سازی کاملی از آن وجود ندارد، و همچنان به عنوان یک ابزار قدرتمند بهینه‌سازی برای حل طیف گسترده‌ای از مسائل در دنیای واقعی به اثبات رسیده است. برخی از کاربردهای برجسته این الگوریتم در بهینه‌سازی مسیر، طراحی مدار، برنامه‌ریزی وسایل نقلیه و غیره است.

در هر مرحله از الگوریتم، مورچگان به دو عملکرد اصلی تکیه می‌کنند: اکتشاف و بهره‌برداری [7].

به طور کلی، الگوریتم کلونی مورچگان شامل مراحل زیر است:

۱. مقداردهی اولیه: مقداردهی اولیه به مورچگان و فرومون‌ها

صورت می‌گیرد.

۲. حرکت مورچگان: مورچگان به صورت تصادفی در فضای جستجو حرکت می‌کنند.

۳. ترکیب فرومون‌ها: مقدار فرومون‌ها بر روی مسیرهایی که مورچگان طی کرده‌اند، به روزرسانی می‌شود.

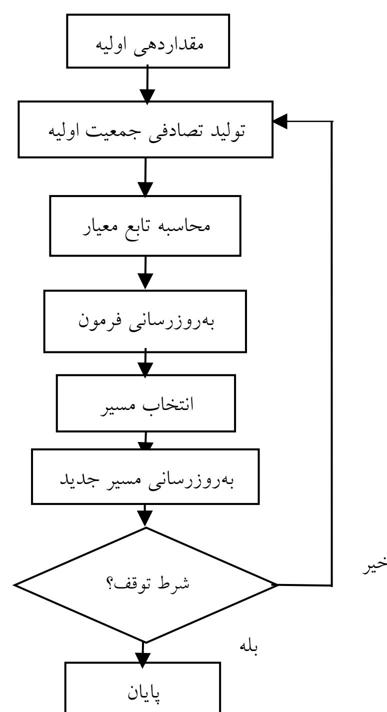
۴. انتخاب مسیر: مورچگان بر اساس مقدار فرومون‌ها و احتمال‌های مرتبط با مسیرها، مسیرهای بهینه را انتخاب می‌کنند.

۵. به‌روزرسانی بهترین مسیر: بهترین مسیرهایی که تا کنون توسط مورچگان پیدا شده‌اند، به عنوان بهترین حل موجود در نظر گرفته می‌شود و به‌روزرسانی می‌شوند.

۶. تا زمان اتمام شرایط خاتمه: مراحل ۲ تا ۵ تکرار می‌شوند تا به شرایط خاتمه برسند [7].

روندنمای الگوریتم کلونی مورچگان

در شکل (۲) نیز روندنمای کلی الگوریتم کلونی مورچگان مشاهده می‌شود.



شکل ۲ روندنمای الگوریتم بهینه‌سازی کلونی مورچگان [30]

معرفی الگوریتم بهینه‌سازی کرم شب‌تاب

الگوریتم فراابتکاری کرم شب‌تاب بر اساس مکانیزم‌های تعاملی کرم‌های شب‌تاب در جستجوی منابع نور الهام گرفته شده است.

در این الگوریتم، کرم‌های شب‌تاب به صورت تصادفی در فضای جستجو قرار می‌گیرند و منابع نور را توزیع می‌کنند. هر کرم شب‌تاب قادر به تشخیص میزان نور در اطراف خود است و با توجه به میزان نور واکنش‌هایی نشان می‌دهد [22].

به طور کلی، الگوریتم کرم شب‌تاب شامل مراحل زیر است:

۱. مقداردهی اولیه: مقداردهی اولیه به کرم‌های شب‌تاب و منابع نور صورت می‌گیرد.

۲. حرکت کرم‌ها: کرم‌های شب‌تاب به‌صورت تصادفی در فضای جستجو حرکت کرده و نور در اطراف خود را اندازه‌گیری می‌کنند.

۳. ارتباطات کرم‌ها: کرم‌ها با توجه به میزان نور در اطراف خود و بر اساس قوانین تعاملی، واکنش‌هایی نشان می‌دهند. این واکنش‌ها شامل جذب منابع نور و جلوگیری از رسیدن کرم‌های دیگر به منابع نور است.

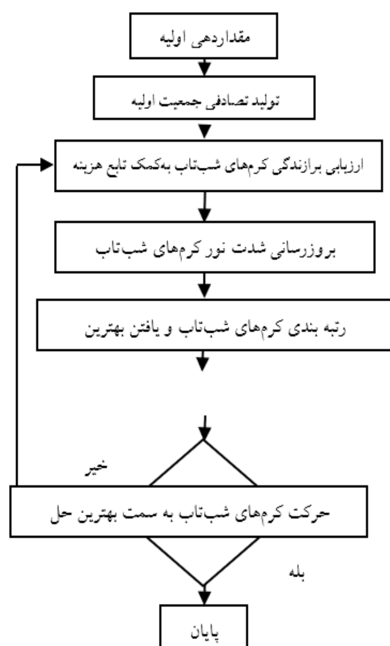
۴. به‌روزرسانی شعاع نور: بر اساس واکنش‌ها و میزان نور در اطراف، شعاع نور هر کرم شب‌تاب به‌روزرسانی می‌شود.

۵. به‌روزرسانی منابع نور: بر اساس شعاع نور کرم‌ها و توزیع آن‌ها در فضای جستجو، منابع نور به‌روزرسانی می‌شوند.

۶. تا زمان اتمام شرایط خاتمه: مراحل ۲ تا ۵ تکرار می‌شوند تا به شرایط خاتمه مانند تعداد تکرارها یا دستیابی به یک حل بهینه برسد [22].

روندنمای الگوریتم کرم شب‌تاب

در شکل (۳) روندنمای الگوریتم کرم شب‌تاب را مشاهده می‌کنیم.



شکل ۳ فلوچارت الگوریتم بهینه‌سازی کرم شب‌تاب [23]

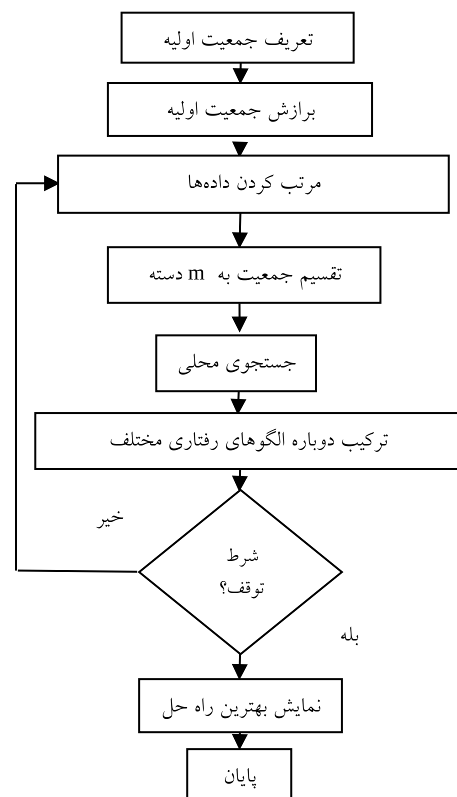
معرفی الگوریتم بهینه سازی قورباغه

الگوریتم قورباغه یک الگوریتم بهینه‌سازی جهش قورباغه، یکی ازالگوریتم‌های بهینه‌سازی فراابتکاری است که از رفتار اجتماعی قورباغه‌ها در طبیعت الهام گرفته شده است [16-14].

الگوریتم بهینه‌سازی قورباغه الگوریتم فراابتکاری است که بر پایه رفتار قورباغه‌ها در جستجوی غذا توسعه یافته است. این الگوریتم برای حل مسائل بهینه‌سازی مورد استفاده قرار می‌گیرد. در این الگوریتم، یک جمعیت از قورباغه‌ها (که هر قورباغه یک حالت ممکن را نشان می‌دهد) ایجاد می‌شود. این قورباغه‌ها در جستجوی یافتن یک حل بهینه برای مسئله قرار می‌گیرند. الگوریتم قورباغه به دلیل رفتار مشابه قورباغه در حرکت و جستجو در فضای سه‌بعدی، برای حل مسائل پیچیده و چندمعیاره کارآمد است. این الگوریتم می‌تواند در مسائل بهینه‌سازی، هوش مصنوعی، بهینه‌سازی ترکیبی و سایر حوزه‌های مرتبط مورد استفاده قرار گیرد [9].

روندنمای الگوریتم قورباغه

در شکل (۴) روندنمای الگوریتم قورباغه را مشاهده می‌کنیم.



شکل ۴ فلوچارت الگوریتم بهینه‌سازی قورباغه [14]

روند کار الگوریتم قورباغه به شرح زیر است:

۱. اولیه: جمعیتی از قورباغه‌ها به صورت تصادفی ایجاد می‌شود.
۲. ترکیب: قورباغه‌ها بر اساس مقدار ارزیابی، به صورت تصادفی در گروه‌های کوچک‌تری تقسیم می‌شوند. گروه‌ها با توجه به مقدار ارزیابی از بهترین تا بدترین مرتب می‌شوند.
۳. بهبود: در هر گروه، قورباغه‌ها با استفاده از روش بهبود محلی وضعیت خود را بهبود می‌بخشند.
۴. حرکت: قورباغه‌ها در هر گروه به صورت تصادفی حرکت می‌کنند و وضعیت جدیدی را برای خود ایجاد می‌کنند.
۵. ادغام: پس از حرکت، گروه‌های قورباغه‌ها ادغام می‌شوند و جمعیت جدیدی تشکیل می‌دهند.
۶. به‌روزرسانی: با توجه به مقدار ارزیابی، قورباغه‌ها به‌روزرسانی می‌شوند و بهترین حالت‌ها دریافت می‌شوند.
۷. تا زمان اتمام شرایط خاتمه: مراحل ۳ تا ۷ تکرار می‌شوند تا به شرایط خاتمه برسیم.
۸. ارزیابی: هر قورباغه وضعیت خود را ارزیابی کرده و مقدار مناسبی برای آن اختصاص می‌دهد [15,16].

معرفی الگوریتم بهینه‌سازی رقابت استعماری

الگوریتم بهینه‌سازی رقابت استعماری یک الگوریتم محاسباتی است که بر اساس مفهوم رقابت و هم‌اندیشی در یک سازمان استعماری تشکیل شده است [13,27] این الگوریتم برای حل مسائل بهینه‌سازی پیچیده و چندمعیاره استفاده می‌شود. الگوریتم بهینه‌سازی رقابت استعماری از ترکیب الگوریتم‌های تکاملی و الگوریتم‌های مبتنی بر رقابت بین گروه‌ها بهره می‌برد [12].

روند کار الگوریتم بهینه‌سازی رقابت استعماری به شرح زیر است:

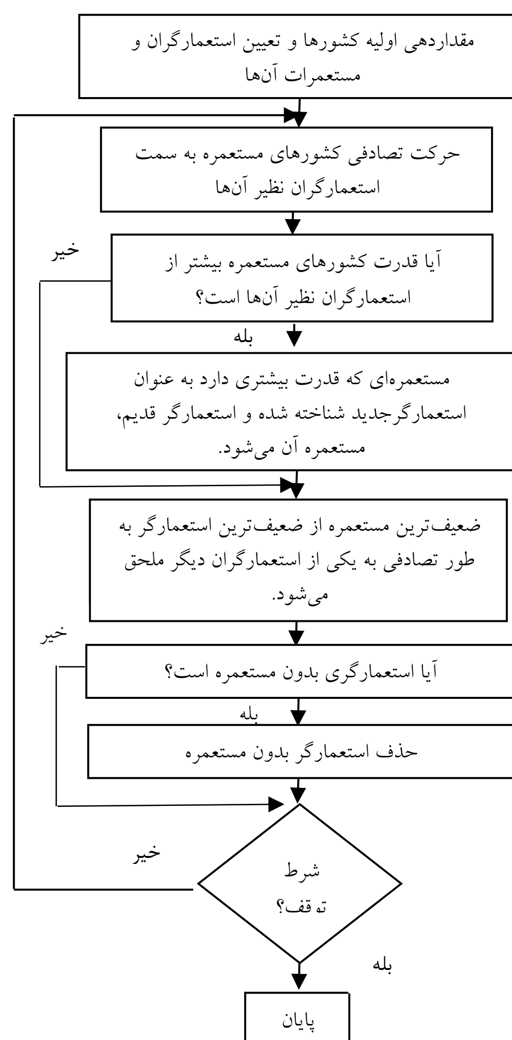
۱. مقداردهی اولیه: استعمارها و جانداران به صورت تصادفی در فضای جستجو قرار می‌گیرند.
۲. ارزیابی: هر جانداران وضعیت خود را ارزیابی می‌کند و مقدار مناسبی برای آن اختصاص می‌دهد.
۳. تعامل استعمارها: استعمارها به صورت مکرر اطلاعات و تجربیات خود را با استفاده از عملکرد جانداران مبادله می‌کنند.
۴. رقابت و انتخاب: جانداران درون هر استعمار به رقابت با یکدیگر در جهت بهبود مقدار ارزیابی خود می‌پردازند. جانداران با برتری در رقابت در استعمار خود، به نسل بعدی منتقل می‌شوند.
۵. خاتمه: تا زمان اتمام کار، مراحل ۳ و ۴ تکرار می‌شوند.
۶. خروجی: پس از پایان مراحل الگوریتم، بهترین حل یافت شده

به عنوان خروجی الگوریتم انتخاب می‌شود. پس از پایان الگوریتم، مقدار بهینه‌سازی شده تابع هدف و مقدار بهینه‌یابی شده برای متغیرها به عنوان خروجی الگوریتم ارائه می‌شود.

در واقع الگوریتم بهینه‌سازی رقابت استعماری به دلیل قابلیت تطبیق با مسائل پیچیده و چندمعیاره و قدرت استفاده از رقابت و هم‌اندیشی بین امپراتورها، به عنوان یک روش قوی و مؤثر در حل مسائل بهینه‌سازی مورد استفاده قرار می‌گیرد. الگوریتم بهینه‌سازی رقابت استعماری قابلیت تعمیم و انعطاف‌پذیری دارد و می‌تواند در مسائل مختلفی از جمله بهینه‌سازی تابع، بهینه‌سازی مسئله‌های ترکیبی و بهینه‌سازی پارامترها مورد استفاده قرارگیرد [10].

روندنمای الگوریتم رقابت استعماری

شکل (۵) روندنمای الگوریتم قورباغه را نشان می‌دهد.



شکل ۵ فلوچارت الگوریتم بهینه‌سازی رقابت استعماری [27]

عامل تطبیقی پویا

در واقع عامل تطبیقی پویا به منظور متعادل‌کردن نرخ همگرایی و توانایی جستجوی بهینه مطلق در طول فرایند جستجو معرفی شده است. این عامل توانایی تطبیق سرعت جستجو را در محیط مسئله دارد و با تغییر سرعت جستجو و تطبیق آن با شرایط محیطی، به‌دنبال بهینه‌سازی بهتری می‌گردد. با این تطبیق پویا، الگوریتم‌ها قادر به بهبود نرخ همگرایی و جستجوی بهینه‌تر در فضای جستجوی خود هستند. در این مقاله، با استفاده از تکنیک‌های خروج از بهینه محلی، این الگوریتم‌ها توانایی جستجو در فضای جستجوی گسترده‌تر را دارند و می‌توانند به‌نتایج بهینه مطلق نزدیک شوند.

در الگوریتم گرگ خاکستری، عامل تطبیقی پویا نقش مهمی در بهبود عملکرد الگوریتم ایفا می‌کند. این عامل معمولا برای تطبیق نرخ اکتشاف و بهبود جستجو استفاده می‌شود و بستگی به پارامترهای مرتبط با نرخ اکتشاف و تغییرات محیطی دارد، رابطه (۱) به خوبی نمایانگر این عامل است:

$$a = 2 - t * \left(\frac{(2)}{max_iter}\right) \qquad (۱)$$

در این رابطه، a نرخ اکتشاف است. t زمان فعلی یا تکرار فعلی است و max_iter حداکثر تعداد تکرارها است.

با استفاده رابطه (۱) عامل تطبیقی پویا می‌تواند نرخ اکتشاف را در طول فرایند جستجو تنظیم کند. در ابتدای جستجو این نرخ افزایش و جستجو تصادفی و منجر به خروج از اکسترمم‌های محلی می‌شود و با گذشت زمان و تکرارها، نرخ اکتشاف کاهش می‌یابد و الگوریتم از جستجوی بیشتری در نزدیکی به مکان‌های بهینه استفاده می‌کند. لذا این تطبیق باعث می‌شود الگوریتم بهبود یابد و نتایج بهینه‌تری را تولید کند.

در الگوریتم کلونی مورچگان نیز عامل تطبیقی پویا معمولا برای تطبیق نرخ انتشار فرومون استفاده می‌شود. در این الگوریتم، یکی از روابط کلیدی که توسط عامل تطبیقی پویا تغییر می‌کند، رابطه انتشار فرومون است. این رابطه به شکل زیر بیان شده است:

$$P_clone(i) = P_min + (P_max - P_min) * \frac{(fit(i) - fit_min)}{(fit_max - fit_min)} \qquad (۲)$$

در رابطه(۲)، P_clone(i) انتشار فرومون در جهت i است. P_min و P_max حداقل وحداکثر مقادیر انتشار فرومون هستند.

fit(i) مقدار تابع ارزیابی (fitness function) برای انتشار فرومون است و fit_min و fit_max حداقل و حداکثر مقادیر تابع ارزیابی در میزان انتشار فرومون هستند.

در واقع عامل تطبیقی پویا با تطبیق مقدار P_clone(i) با شرایط محیطی و مقادیر تابع ارزیابی، انتشار فرومون را تنظیم می‌کند. لذا مسیرهایی‌که توسط مورچه‌های بیشتردنبال می‌شوند، با غلظت فرومون بیشتری علامت‌گذاری می‌شوند و از این طریق امکان یافتن بهینه‌ترین حل و هدایت مورچه‌ها به سمت منابع غذایی را فراهم می‌کند.

در الگوریتم کرم شب‌تاب هم، فاکتور تصادفی‌سازی α به‌عنوان عامل پویا در طی فرایند جستجو تعدیل می‌شود. در اینجا k بین ۰ و ۱ است و نرخ افزایشی نام دارد که می‌تواند سرعت کاهش α را تعدیل کند، رابطه (۳) نمایانگر این عامل تطبیقی پویا است:

$$\alpha(t+1) = \alpha(t).\exp\left(-k.\frac{t}{G_{max}}\right) \qquad (۳)$$

نرخ کاهشی k ممکن است روی کارایی پارامتر مدل تأثیر بگذارد. در رابطه (۳) G_{max} حداکثر تعداد تکرار و t نشان دهنده شاخص تولید نور می‌باشد.

در ادامه در الگوریتم قورباغه نیز عامل تطبیقی پویا برای بهبود نرخ همگرایی و جستجو در فضای جستجوی الگوریتم استفاده می‌شود.

در این الگوریتم، از رابطه (۴) برای تغییر سرعت جستجو استفاده می‌شود:

V(t+1) = V(t) + A(t) * (X_best - X(t)) (۴)

در این رابطه، V(t+1) سرعت جستجو در زمان t+1 است.

V(t) سرعت جستجو در زمان t است.

A(t) عامل تطبیقی پویا است که مقدار آن در طول فرایند جستجو تغییر می‌کند. X_best بهترین حل یافت شده تا آن لحظه است.

و X(t) مکان فعلی جستجو در زمان t است.

در حقیقت عامل تطبیقی پویا با تطبیق مقدار A(t) با شرایط محیطی، سرعت جستجو را تنظیم می‌کند. مقدار A(t) می‌تواند به صورت ثابت، تصادفی یا با استفاده از روش‌های خاص تعیین شود. تغییر مقدار A(t) در طول فرایند جستجو، به این الگوریتم قابلیت تطبیق با محیط و بهبود نرخ همگرایی را می‌دهد [28].

در آخر در الگوریتم رقابت استعماری عامل تطبیقی پویا نقش مهمی در تطبیق و بهبود عملکرد الگوریتم ایفا می‌کند. این عامل معمولا برای تطبیق نرخ مهاجرت و تغییر مقدار آن در طول فرایند جستجو استفاده می‌شود. رابطه‌ای که توسط عامل تطبیقی پویا در الگوریتم رقابت استعماری تأثیر دارد، بستگی به پارامترهای مرتبط با مهاجرت و تغییرات محیطی دارد. رابطه (۵) از عامل تطبیقی پویا در این الگوریتم استفاده می‌کند:

Migration_rate(t) = Migration_rate_initial*e^(-alpha*t) (۵)

در این رابطه، Migration_rate(t) نرخ مهاجرت در زمان t است. عبارت Migration_rate_initial نرخ مهاجرت اولیه است. e نماد عدد اویلر است.

alpha پارامتری است که توسط عامل تطبیقی پویا تنظیم می‌شود و t زمان فعلی در الگوریتم است. با استفاده از رابطه (۵) عامل تطبیقی پویا می‌تواند نرخ مهاجرت را در طول فرایند جستجو تنظیم کند. با گذشت زمان، نرخ مهاجرت کم می‌شود و تأثیر مهاجرت بین جمعیت‌ها کاهش می‌یابد. این تطبیق باعث می‌شود که الگوریتم بهبود یابد و به جمعیت‌هایی که در نتیجهٔ عملکرد بهتری هستند، اجازه بدهد که بیشتر منتشر شوند و تأثیر بیشتری در روند جستجو داشته باشند. لذا عامل تطبیقی پویا در الگوریتم‌های تکاملی کلونی مورچگان، قورباغه، گرگ خاکستری، رقابت استعماری و کرم شب‌تاب با افزایش خود در ابتدای جستجو می‌تواند موجب خروج از بهینه محلی، عملکرد بهتر و جستجوی تصادفی و گسترده‌تر در فضای جستجوی الگوریتم‌ها شوند و با کاهش تدریجی به رسیدن به نقاط بهینه کمک کنند.

معرفی مسائل بهینه‌سازی مهندسی

معرفی مسائل بهینه‌سازی مهندسی مقید

در این بخش، پنج مسئله از مسائل بهینه‌سازی مهندسی مقید شده که دارای توابع هدف، متغیرهای طراحی و قیود مختلف هستند، انتخاب می‌شوند و عملکرد الگوریتم کرم شب‌تاب، رقابت استعماری و بقیه الگوریتم‌های پیشنهادی از نظر کیفیت راه‌حل و پایداری و همچنین نرخ هم‌گرایی مورد بررسی قرار می‌گیرد. این پنج مسئله بهینه‌سازی مهندسی مقید با توجه به‌دسته‌بندی

متغیرهایشان به مسائل بهینه‌سازی پیوسته و گسسته تقسیم می-شوند.

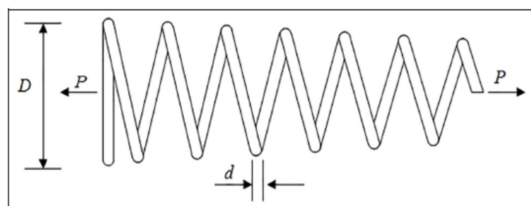
نتایج و بهترین راه‌حل‌های همه الگوریتم‌ها به کمک حساسیت‌سنجی پارامترها، برای این مسائل بهینه‌سازی مهندسی، در بیش از ۳۰ اجرای مستقل به دست می‌آیند. جداول این بخش شامل پنج الگوریتم موجود در بخش روش‌های حل این مقاله و الگوریتم‌های موجود در مرجع [29] است که با رنگ خاکستری در ردیف‌های انتهایی جداول مشخص شده است.

مسائل بهینه‌سازی مقید با متغیرهای پیوسته

مسئله طراحی فنر کششی/ فشردهی. شکل شماره (۶) شماتیک یک فنر کششی/ فشرده را نشان می‌دهد.

هدف طراحی مسئله کشش/ فشردگی به حداقل رساندن وزن آن با اعمال قیود حداقل تغییر شکل، تنش برشی، فرکانس موج و حداکثر قطر خارجی است، به طوری که تغییر شکل در فنرها معمولا به قانون هوک (نسبت نیروی بازگرداننده وارده از سوی فنر به میزان تغییر شکل فنر برابر است با مقدار ثابت فنر یا k) مرتبط است. فرکانس موج نیز معمولا با توجه به ویژگی‌های مکانیکی فنر، از جمله سختی (k) و جرم (m) سیستم محاسبه می‌شود.

در واقع این قیود، چهار معادله نابرابر غیرخطی را تشکیل می‌دهند که در رابطه (۶) شرح داده شده است. مسئله طراحی دارای سه متغیر طراحی شامل قطر سیم d، قطر متوسط سیم‌پیچ D و تعداد سیم پیچ‌های فعال N است که در تابع هدف J(x) و تابع قیود به ترتیب با x1، x2 و x3 نشان داده شده است. در واقع تعداد دور سیم‌پیچ یعنی x3 باید از نوع عدد صحیح انتخاب گردد. تابع برازندگی و یا معیار و محدودیت‌های این مسئله نیز در رابطه (۶) تشریح شده است. در این میان قابل ذکر است که وزن فنر به طور کلی با حجم ماده‌ای که برای ساخت آن استفاده شده است، ارتباط دارد و حجم فنر به قطر سیم، تعداد دورها و طول فنر بستگی دارد. در جدول (۱) الگوریتم کرم شب‌تاب با چندین الگوریتم بهینه‌سازی فراابتکاری دیگر از جمله الگوریتم مورچگان، رقابت استعماری، ازدحام ذرات، گرگ خاکستری و غیره مقایسه شده و همان طور که مشاهده می‌شود الگوریتم کرم شب‌تاب پیشنهادی بهترین حل را با مقدار تابع هدف ۰/۰۱۲۶ ارائه می‌دهد.



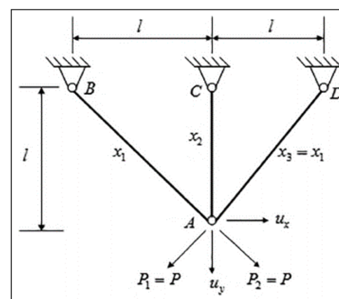
شکل ۶ شماتیک فنرکششی [30]

جدول ۱ مسئله طراحی فنر کششی فشاری [29]

الگوریتم	x_1	x_2	x_3	g_1	g_2	g_3	g_4	J
ICA	۰/۰۵۶۹	۰/۴۹۶۳	۶/۱۷۱۱	−۰/۱۰۵۳	−۰/۴۴۱۹	−۱/۵۸۴۲	−۰/۶۰۳۰	۰/۰۱۳۱
ACOR	۰/۰۵۷۰	۰/۴۹۷۷	۶/۱۳۴۰	−۱/۰E-۴	−۱/۰E-۴	−۴/۲۶۶۴	−۰/۶۳۰۲	۰/۰۱۳۱
SFLA	۰/۰۵۰۰	۰/۳۱۵۷	۱۴/۲۸۶۳	−۰/۲۱۷۸	−۰/۶۳۹۹	−۰/۵۷۳۱	−۰/۳۲۳۹	۰/۰۱۲۸
GWO	۰/۰۵۲۷	۰/۳۸۲۱	۹/۹۶۷۹	−۵/۲۴E-۴	−۷/۵۶E-۴	−۴/۰۲۳۶	−۵/۰۰۵۲	۰/۰۱۲۸
FA	۰/۰۵۱۰	۰/۳۳۹۴	۱۲/۳۸۵۸	−۱/۴۴E-۱۰	−۱/۵۱E-۱۱	−۴/۰۰۳۹	−۰/۵۵۱۸	**۰/۰۱۲۶**
IAPSO	۰/۰۵۱۶	۰/۳۵۶۶	۱۱/۲۹۴۱	−۱/۹۷E-۱۰	−۴/۶۴E-۱۰	−۴/۰۵۳۶	−۱/۰۹۱۶	۰/۰۱۲۶
MBA	۰/۰۵۱۶	۰/۳۵۵۹	۱۱/۳۴۴۶	−۰/۰۰۰۹	−۰/۱۳۴۴	−۴/۰۵۲۲	−۰/۷۲۸۲	۰/۰۱۲۶
DAPSO-GA	۰/۰۵۰۰	۰/۳۷۴۴	۸/۵۴۶۵	−۱/۲۵E-۸	−۱/۴۴E-۷	−۴/۸۶۰۷	−۰/۷۱۷۰	۰/۰۰۹۸

IAPSO: improved adaptive particle swarm optimization; MBA: mine blast algorithm DAPSO-GA: dynamic adaptive particle swarm optimization and genetic algorithm

مسئله طراحی خرپای سه‌میله متقارن. شکل (۷) نمودار شماتیک یک سازه خرپایی متقارن سه‌میله‌ای را نشان می‌دهد. سازه خرپایی متقارن سه‌میله‌ای از فولاد تشکیل شده و تحت دو بار ثابت P1=P2=P قرار دارد. هدف مسئله، طراحی بهینه سازه خرپای سه‌میله‌ای و به حداقل رساندن حجم در معرض قیود تنش است. متغیرهای طراحی، سطح مقطع سه میله x1، x2 و x3 هستند.



شکل ۷ شماتیک یک سازه خرپایی متقارن سه میله ای [29]

تابع هزینه و محدودیت‌های این مسئله در رابطه (۷) آورده شده‌اند.

الگوریتم‌های بخش روش‌های حل این مقاله درکنار الگوریتم‌های مرجع [29] با اندازه ازدحام ۲۰ و حداکثر تعداد تکرار ۱۰۰ برای حل این مسئله بهینه سازی استفاده می‌شود.

همانطور که مشهود است الگوریتم DAPSO-GA از مرجع [29] حل بهتری را در مقایسه با سایر الگوریتم‌ها ارائه می‌دهند. درجدول (۲) تمامی الگوریتم بهینه‌سازی فراابتکاری در کنار دیگر مقایسه شده و همانطورکه مشاهده می‌شود الگوریتم قورباغه پیشنهادی بهترین حل را با مقدار تابع هدف ۲۶۳/۸۰ ارائه می‌دهد.

$$x = [x_1 \ \ x_2 \ \ x_3] = [d \ \ D \ \ N]$$

$$0.05 \le x_1 \le 2.0,\ 0.25 \le x_2 \le 1.3,\ 2.0 \le x_3 \le 15.0$$

$$J(x) = (x_3 + 2)x_2x_1^2$$

$$g_1(x): 1 - \frac{x_2^3 x_3}{71785 x_1^4} \le 0.$$

$$g_2(x): \frac{4x_2^2 - x_1x_2}{12566(x_2x_1^3 - x_1^4)} + \frac{1}{5108x_1^2} - 1 \le 0.$$

$$g_3(x): 1 - \frac{140.45x_1}{x_2^2 x_3} \le 0.$$

$$g_4(x): \frac{x_1 + x_2}{1.5} - 1 \le 0$$

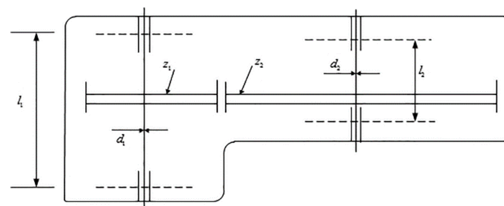
(۶)

مسئله طراحی کاهنده سرعت. در شکل (۸) تصویر شماتیک یک کاهنده سرعت نشان داده شده است. طرح بهینه‌سازی طراحی کاهش دهنده سرعت، برای به حداقل رساندن وزن آن، با توجه به قیود استحکام مربوط به تنش خمشی دندانه‌های چرخ‌دنده و تنش سطحی و تنش‌ها و انحراف عرضی شفت‌ها است. متغیرهای طراحی این مسئله طراحی شامل عرض (b)، مدول دندانه (m)، تعداد دندانه‌ها در پینیون (z)، طول شفت اول و دوم بین یاتاقان‌های آن‌ها (l1 و l2)، قطر شفت اول (d1) و قطر شفت دوم (d2) است. این متغیرهای طراحی به ترتیب با x1، x2، x3، x4، x5، x6 و x7 در معادلات تابع هدف و قیود نشان داده می‌شوند. متغیر طراحی x3 (یعنی تعداد دندانه در پینیون) یک متغیر طراحی گسسته است و بقیه متغیرها پیوسته هستند. همچنین تابع برازندگی و یا معیار و محدودیت‌های این مسئله در رابطه (۸) آورده شده‌اند.

جدول ۲ مسئله طراحی خرپای سه‌میله‌ای [29]

الگوریتم	x_1	x_2	g_1	g_2	g_3	J
ACOR	۰/۸۰۸۳	۰/۳۵۵۳	−۱/۰E-۵	−۱/۵۲۵۸	−۱/۰۵۶۲	۲۶۴/۱۶
GWO	۰/۷۹۷۵	۰/۳۸۳۸	−۰/۶۰۱۲	−۱/۱۶۲۳	−۰/۶۲۳۱	۲۶۳/۹۵
FA	۰/۷۹۶۴	۰/۳۸۶۸	−۱/۳۱E-۹	−۶/۵۵۱۲	−۰/۵۱۲۳	۲۶۳/۹۳
ICA	۰/۷۸۳۳	۰/۴۲۳۸	−۰/۴۳۳۴	−۱/۴۳۳۹	−۰/۹۹۹۶	۲۶۳/۹۲
SFLA	۰/۷۸۵۶	۰/۴۱۷۱	−۰/۱۷۰۲	−۱/۳۷۹۶	−۰/۷۹۰۶	**۲۶۳/۸۰**
PSO-TVAC	۰/۷۸۸۷	۰/۴۰۸۱	−۴/۴۴E-۱۳	−۱/۴۶۴۲	−۰/۵۳۵۸	۲۶۳/۸۹
SC	۰/۷۸۸۶	۰/۴۰۸۴	−۸/۲۷E-۹	−۱/۴۶۳۹	−۰/۵۳۶۱	۲۶۳/۸۹
DAPSO-GA	۰/۷۸۸۶	۰/۴۰۸۲	−۲/۳۱E-۰۹	−۱/۴۶۴۱	−۰/۵۳۵۸	**۲۶۳/۸۹**

PSO-TVAC: Particle swarm optimization withtime-varying accelerating coefficients. ;SC :Society and civilization algorithm; DAPSO-GA: dynamic adaptive particle swarm optimization and genetic algorithm.



شکل ۸ شماتیک یک کاهنده سرعت [30]

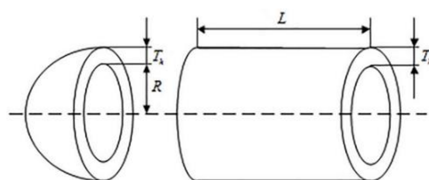
الگوریتم‌های بخش روش‌های حل درکنار الگوریتم‌های مرجع [29] با سایز ازدحام ۴۰ و حداکثر تعداد تکرارهای ۵۰۰ برای حل این مسئله بهینه‌سازی استفاده می‌شود.

$$2.6 \le x_1 \le 3.6,\ 0.7 \le x_2 \le 0.8,\ 17 \le x_3 \le 28,$$
$$7.3 \le x_4, x_5 \le 8.3, 2.9 \le x_6 \le 3.9, 5.0 \le x_7 \le 5.5$$
$$J(x) = 0.7854x_1x_2^2(3.3333x_3^2 + 14.9334x_3 - 43.0934)$$
$$-1.508x_1(x_6^2 + x_7^2) + 7.4777(x_6^3 + x_7^3)$$
$$+0.7854(x_4x_6^2 + x_5x_7^2)$$
$$g_1(x): \frac{27}{x_1x_2^2x_3} - 1 \le 0. \quad g_2(x): \frac{397.5}{x_1x_2^2x_3^2} - 1 \le 0.$$
$$g_3(x): \frac{1.93x_4^3}{x_2x_6^4x_3} - 1 \le 0. \quad g_4(x): \frac{1.93x_5^3}{x_2x_7^4x_3} - 1 \le 0.$$
$$g_5(x): \frac{[(745\,x_4/x_2x_3)^2 + 16.9 \times 10^6]^{0.5}}{110x_6^3} - 1. \le 0$$
$$g_6(x): \frac{[(745\,x_6/x_2x_3)^2 + 157.5 \times 10^6]^{0.5}}{110x_7^3} - 1. \le 0.$$

(۸)

مقدار تابع هدف در تکرارهای اولیه به سرعت به حداقل نزدیک می‌شود (کمتر از ۵۰ تکرار)، که نرخ همگرایی بالای این الگوریتم را برای این مسئله طراحی گسسته مقید نشان می‌دهد. همان طور که نشان داده شده است، الگوریتم DAPSO-GA از مرجع [29] حل مناسبی را در مقایسه با سایر الگوریتم‌ها ارائه می‌دهند. در جدول (۳) الگوریتم کرم شب‌تاب نیز با الگوریتم بهینه‌سازی فراابتکاری دیگر مقایسه شده و همان طور که مشاهده می‌شود الگوریتم رقابت استعماری پیشنهادی بهترین حل را با مقدار تابع هدف ۲۹۹۰/۰۴ ارائه می‌دهد.

مسئله طراحی مخزن تحت فشار. شکل (۹) شماتیکی از یک مخزن تحت فشار را نشان می‌دهد. هدف طراحی به حداقل رساندن هزینه ساخت کل آن، از جمله هزینه‌های مواد، شکل‌دهی و جوش است. متغیرهای طراحی شامل ضخامت پوسته Ts، ضخامت سر Th، شعاع داخلی R و طول مقطع استوانه‌ای ظرف L، که در آن Ts و Th متغیرهای گسسته و مضرب صحیح ۰،۰۶۲۵ هستند، درحالی که R و L متغیرهای پیوسته هستند. این چهار متغیر طراحی به ترتیب با x1، x2، x3 و x4 در تابع هدف اعمال شده و معادلات قیود هم در رابطه (۹) ارائه شده است.



شکل ۹ شماتیک از یک مخزن تحت فشار [30]

جدول ۳ مسئله طراحی کاهنده سرعت [29]

الگوریتم	x1	x2	x3	x4	x5	x6	x7	J
GWO	۳/۵۱۹۸	۰/۷	۱۷	۷/۵۴۳۱	۸/۰۱۷۷	۳/۴۲۵۴	۵/۲۸۸۹	۳۰۳۲/۳۶
FA	۳/۵۳۴۴	۰/۷	۱۷	۷/۸۲۱۷	۸/۰۴۲۱	۳/۳۵۱۲	۵/۲۸۶۸	۳۰۲۰/۰۶
SFLA	۳/۵۰۰۰	۰/۷	۱۷	۷/۴۵۱۸	۷/۹۰۱۱	۳/۳۵۰۵	۵/۲۹۴۴	۳۰۰۴/۹۳
ACOR	۳/۵۰۳۰	۰/۷	۱۷	۷/۴۰۶۹	۷/۸۲۱۶	۳/۳۵۳۴	۵/۲۸۶۷	۳۰۰۴/۰۴
ICA	۳/۵۰۰۱	۰/۷	۱۷	۷/۳۰۰۲	۷/۷۱۵۷	۳/۳۵۰۲	۵/۲۸۶۷	**۲۹۹۰/۰۴**
IAPSO	۳/۵۰۰۰	۰/۷	۱۷	۷/۳۰۰۰	۷/۷۱۵۳	۳/۳۵۰۲	۵/۲۸۶۶	۲۹۹۴/۴۷
MBA	۳/۵۰۰۰	۰/۷	۱۷	۷/۳۰۰۰	۷/۷۱۵۷	۳/۳۵۰۲	۵/۲۸۶۶	۲۹۹۴/۴۸
DAPSO-GA	۳/۵۰۰۰	۰/۷	۱۷	۷/۳۰۰۰	۷/۷۱۵۳	۳/۳۵۰۲	۵/۲۸۶۶	**۲۹۹۴/۴۷**

IAPSO: improved adaptive particle swarm optimization; MBA: mine blast algorithm DAPSO-GA: dynamic adaptive particle swarm optimization and genetic algorithm

جدول ۴ مسئله طراحی مخزن تحت فشار [29]

الگوریتم	x_1	x_2	x_3	x_4	g_1	g_2	g_3	g_4	J
ACOR	۰/۸۴۶۲	۰/۴۱۷۸	۴۳/۸۹	۱۵۵/۵۶	۱/۰E-۳	۱/۰E-۳	۱/۰E-۳	−۸۴/۴۳	۶۰۰۴/۰۹
FA	۰/۸۰۰۷	۰/۳۹۷۱	۴۱/۳۷	۱۸۶/۰۴	−۱/۴E-۹	−۱/۵E-۴	−۰/۸E-۶	−۵۹/۵	۵۹۴۹/۰۶
ICA	۰/۷۸۴۲	۰/۳۸۵۴	۴۰/۵۳	۱۹۷/۰۷	−۰/۰۳۱۵	−۵/۲۶۹۷	−۲/۹۶E۴	−۱۳۰/۸۰	۵۹۰۳/۰۱
SFLA	۰/۷۸۰۶	۰/۳۸۴۸	۴۰/۴۳	۱۹۸/۵۶	−۰/۰۹۰۵	−۱/۷۱۷۷	−۳/۶E۴	−۱۱۱/۳۱	۵۸۹۱/۳۰
GWO	۰/۷۷۸۳	۰/۳۸۶۱	۴۰/۳۲	۱۹۹/۹۶	−۶/۲E-۱۵	−۱/۱E-۱۶	−۷۹/۹۹	−۴۴/۳	**۵۸۹۰/۷۵**
IAPSO	۰/۸۱۲۵	۰/۴۳۷۵	۴۲/۰۹	۱۷۶/۶۳	−۴/۰E-۱۳	−۳/۵E-۰۲	−۱/۳E-۰۷	−۶۳/۳	۶۰۵۹/۷۱
MBA	۰/۷۸۰۲	۰/۳۸۵۶	۴۰/۴۲	۱۹۸/۴۷	۰	۰	−۸۶/۳۶	−۴۱/۵	۵۸۸۹/۳۲
DAPSO-GA	۰/۸۱۲۵	۰/۴۳۷۵	۴۲/۰۹	۱۷۶/۶۳	−۴/۰E-۱۳	−۳/۵E-۰۲	−۱/۳E-۷	−۶۳/۳	**۶۰۵۹/۷۱**

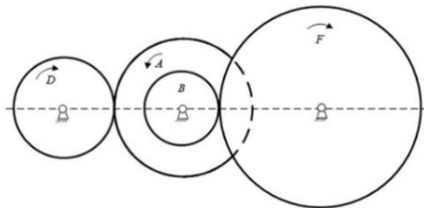
IAPSO: improved adaptive particle swarm optimization; MBA: mine blast algorithm DAPSO-GA: dynamic adaptive particle swarm optimization and genetic algorithm

الگوریتم کرم شب‌تاب پیشنهادی با اندازه ازدحام ۲۵ و حداکثر تعداد تکرار ۵۰۰ عدد، برای حل این مسئله بهینه‌سازی استفاده می‌شود. همان طور که مشهود است. الگوریتم MBA از مرجع [29] حل بهتری را در مقایسه با سایر الگوریتم‌ها ارائه می‌دهند. در جدول (۴) تمامی الگوریتم بهینه‌سازی فراابتکاری در کنار یکدیگر مقایسه شده و همان طور که مشاهده می‌شود الگوریتم گرگ خاکستری پیشنهادی بهترین حل را با مقدار تابع هدف ۵۸۹۰/۷۵ پیدا می‌کند.

$$x = [\,x_1\ \ x_2\ \ x_3\ \ x_4\,] = [\,T_s\ \ T_h\ R\ \ L\,]$$
$$x_1\ \geq 1 \times 0.0625,\ \ x_2\ \leq 99 \times 0.0625,\ \ x_3\ \geq 10.,$$
$$x_4 \leq 200.$$
$$J(x) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.166x_1^2x_4 + 19.84x_1^2x_3$$
$$g_1(x):-x_1 + 0.0193x_3\ \leq 0.$$
$$g_2(x):-x_2 + 0.00954x_3 \leq 0.$$
$$g_3(x):-\pi x_3^2 - \frac{4}{3}\pi x_3^3 + 1296000 \leq 0.$$
$$g_4(x): x_4 - 240 \leq\ 0. \qquad (۹)$$

مسائل بهینه‌سازی مهندسی مقید با متغیرهای گسسته [29]

مسئله طراحی مجموعه چرخ‌دنده‌ای. شکل (۱۰) یک نمودار شماتیک از یک مجموعه دنده‌ای را نشان می‌دهد که از چهارچرخ دنده تشکیل شده است. طرح مسئله بهینه‌سازی طراحی مجموعه چرخ‌دنده‌ای، جهت به‌حداقل‌رساندن خطا بین نسبت دنده به‌دست‌آمده و نسبت دنده مورد نیاز، یعنی۱ برروی ۶/۹۳۱ بوده که برروی قیود که تعداد دندانه‌های چهار چرخ‌دنده است و فقط در محدوده مجاز متغیرهای طراحی است، اعمال می‌شود. این مسئله یک مسئله بهینه‌سازی گسسته است، زیرا همه متغیرهای طراحی اعداد صحیح هستند. تعداد دندانه‌های چرخ‌دنده‌های A، B، D و F (یعنی متغیرهای طراحی) در رابطه (۵) به ترتیب با x1، x2، x3 و x4 در تابع هدف، نشان داده شده است. تابع برازندگی و یا معیار و محدودیت‌های این مسئله نیز در رابطه (۵) آورده شده‌اند. برای حل این مسئله بهینه‌سازی از الگوریتم کرم شب‌تاب گسسته با اندازه ازدحام ۱۰۰ و حداکثر تعداد تکرار ۵۰۰ استفاده شده است. همان طور که نشان داده شده است الگوریتم DAPSO-GA از مرجع [30] حل مناسبی را در مقایسه با سایر الگوریتم‌ها برای مسئله طراحی مجموعه چرخ‌دنده‌ای ارائه می‌دهند. جدول (۵) مقایسه راه‌حل‌های بهینه ارائه شده توسط روش‌های موجود در بخش روش‌های حل و الگوریتم‌های مرجع [30] را ارائه می‌دهد. ملاحظه می‌شود که الگوریتم رقابت استعماری پیشنهادی بهترین حل را با مقدار تابع هدف ۲/۷E-۳۴ ارائه می‌دهد.



شکل ۱۰ شماتیک از یک مجموعه چرخ‌دنده‌ای [29]

$$x = [x_1\ \ x_2\ \ x_3\ \ x_4] = [N_1\ \ N_2\ \ N_3\ \ N_4]$$
$$12\ \leq\ x_I\ \leq 60\,, i = 1, 2, 3, 4$$
$$J(x) = \left(\frac{1}{6.931} - \frac{x_2x_3}{x_1x_4}\right)^2 \qquad (۵)$$

جدول ۵ مسئله طراحی مجموعه چرخ‌دنده‌ای

الگوریتم	x_1	x_2	x_3	x_4	J
FA	۵۹	۱۴	۱۴	۲۳	۲/۴۶E-۸
GWO	۵۹	۲۲	۱۲	۳۱	۳/۸۲E-۹
ACOR	۵۰	۲۱	۱۴	۴۳	۱/۸۸E-۱۲
SFLA	۵۷	۳۳	۱۴	۵۶	۳/۰۸E-۳۳
ICA	۴۵	۲۲	۱۳	۴۵	۷/۷۰E-۳۴
IAPSO	۴۳	۱۶	۱۹	۴۹	۲/۷۰E-۱۲
MBA	۴۳	۱۶	۱۹	۴۹	۲/۷۰E-۱۲
DAPSO-GA	۴۹	۱۶	۱۹	۴۳	۲/۷E-۱۲

IAPSO: improved adaptive particle swarm optimization; MBA: mine blast algorithm DAPSO-GA: dynamic adaptive particle swarm optimization and genetic algorithm

نتایج

آزمایش حساسیت‌سنجی

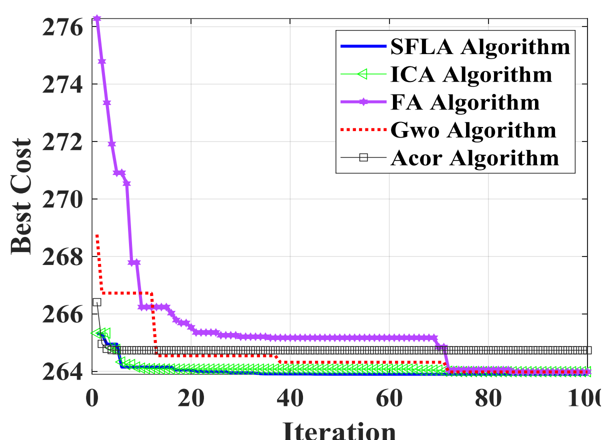
به طور کلی معیار اعداد همگرایی در هر الگوریتم فراابتکاری به طور مجزا، نشان دهنده تعداد مراحل مورد نیاز برای رسیدن به بهینه مطلق است.

هر چه عدد همگرایی کمتر باشد، الگوریتم بهتر و سریعتر به بهینه مطلق همگرا می‌شود. لذا در این مقاله به کمک آزمون حساسیت‌سنجی روی پارامترهای مهم الگوریتم‌های فراابتکاری و بررسی حساسیت آن‌ها نسبت به تغییر در هر یک از این پارامترها، بهترین تنظیمات پارامترها برای هر الگوریتم بهینه‌سازی جهت دستیابی به عدد همگرایی بهینه در حل تمام مسائل تعیین شده است.

به عنوان مثال در الگوریتم فراابتکاری قورباغه برای حل مسئله طراحی خرپای سه‌میله‌ای، پارامترهای بیشینه تعداد تکرار، جمعیت اولیه قورباغه‌ها، تعداد ممپلکس‌ها (memeplex)، اندازه ممپلکس‌ها در آزمایش حساسیت‌سنجی مورد ارزیابی قرار گرفته و نتیجه نهایی حساسیت‌سنجی الگوریتم فراابتکاری قورباغه در شکل (۱۱) در کنار بقیه الگوریتم‌های فراابتکاری مشاهده می-گردد.

نتایج همگرایی

شکل (۱۱) تاریخچه همگرایی تابع برازندگی مسئله طراحی خرپای سه‌میله‌ای برای پنج الگوریتم بهینه‌سازی فراابتکاری را نشان می‌دهد.



شکل ۱۱ تاریخچه همگرایی تابع هزینه مسئله طراحی خرپای سه‌میله‌ای توسط پنج الگوریتم بهینه‌سازی فراابتکاری

و نتایج تأییدکننده موفقیت الگوریتم قورباغه برای مسئله طراحی خرپای سه‌میله‌ای است.

نمایش سرعت اجرا در حل الگوریتم‌ها

در این بخش مطابق جدول (۶) برای مسئله خرپای سه‌میله‌ای، پنج الگوریتم فراابتکاری از نظر سرعت اجرا، با مقدار برازندگی مشخص و در ۱۰۰ تکرار مورد ارزیابی قرار گرفته است.

در پایان نیز پیشنهاد می‌گردد روش‌های بررسی شده، با الگوریتم‌های کلاسیک، روش‌های مبتنی بر شبکه‌های عصبی و غیره مقایسه شده و ترکیب این روش‌ها نیز برای حل مسائل متنوع در دیگر حوزه‌ها و مسائل مهندسی مورد ارزیابی قرار گیرد.

جدول ۶ نتیجه آزمون میانگین زمان اجرای الگوریتم‌ها

زمان اجرا به ثانیه	نوع الگوریتم	
۱۸	گرگ خاکستری	۱
۲۴	رقابت استعماری	۲
۲۵	کرم شب‌تاب	۳
۴۰	قورباغه	۴
۴۲	کلونی مورچگان	۵

نتیجه‌گیری

در این مقاله پنج مسئله در حوزه مهندسی مکانیک به همراه قیود و نتایج مربوطه مورد بررسی قرار گرفته است. الگوریتم‌های گرگ خاکستری، الگوریتم کلونی مورچگان، الگوریتم کرم شب‌تاب، رقابت استعماری و قورباغه در حل این مسائل مورد ارزیابی قرار گرفته‌اند و نتایج مربوطه شرح داده شده است. از نوآوری‌های این مقاله، ارضای قیود مسائل است که در ابتدا به علت محدود بودن بازه جستجو و ایجاد همسایگی تصادفی در ناحیه‌ای که قیود و محدودیت‌ها را ارضا می‌کند، انجام می‌شود. نکته دیگر استفاده از عامل تطبیقی پویا، در تمامی این الگوریتم‌ها است، به طوری که با تغییر سرعت جستجو و تطبیق آن با شرایط محیطی، منجر به بهینه‌سازی بهتر می‌گردد.

نتایج این الگوریتم‌ها به همراه نتایج سایر محققان در جداول (۱) تا (۵) مقایسه شده‌اند.

به عنوان نمونه، نتایج حل مسئله طراحی خرپای سه‌میله‌ای توسط پنج الگوریتم بهینه‌سازی فراابتکاری نشان می‌دهند که الگوریتم قورباغه، با دقت ۰/۱۳۶ درصد در این مسئله جواب دقیق‌تری دارد. در حقیقت برای مسائل مختلف با توجه به گسسته یا پیوسته بودن مسئله، یک الگوریتم بهینه‌سازی مشخص، برای هر یک از مسائل پاسخ دقیق‌تری ارائه می‌دهد. این نتایج نشان می‌دهند که الگوریتم‌های فوق در مسائل مختلف می‌توانند همزمان دقت و سرعت را بهبود بخشیده و این از مظاهر هوش ازدحامی است.

واژه نامه

الگوریتم‌های فراابتکاری	Metaheuristic algorithms
عامل تطبیقی پویا	Dynamic Adaptive Agent
گرگ خاکستری	Gray Wolf
کلونی مورچگان	Ant Colony
کرم شب‌تاب	Firefly
رقابت استعماری و	Colonial Competitive
قورباغه	Algorithm

تقدیر و تشکر

مراجع

[1] N. B. Guedria, “Improved accelerated PSO algorithmform echanical engineering optimization problems,” *Applied Soft Computing*, vol. 40, pp. 455–467, 2016. https://doi.org/10.1016/j.asoc.2015.10.048.

[2] J. Kennedy, R. Eberhart, “Particle swarm optimization,” In Proceedings of ICNN'95-international conference on neural networks, vol. 4, 1995, pp. 1942-1948. https://doi.org/10.1109/ICNN.1995.488968.

[3] J. H. Holland, “Outline for a logical theory of adaptive systems,” Journal of the ACM (JACM), vol. 9, no. 3, pp. 297-314, 1962. https://doi.org/10.1145/321127.321128.

[4] J. D. Bagley, “The behavior of adaptive systems which employ genetic and correlation algorithms,” Ann Arbor, MI: Dissertation Abstracts International, University of Michigan, 1967.

[5] S. Mirjalili, S. M. Mirjalili and A. Lewis, “Grey wolf optimizer,” *Advances in engineering software*, vol. 69, pp. 46-61, 2014. https://doi.org/10.1016/j.advengsoft.2013.12.007.

[6] Y. Li, W. Sun and S. Guan, “A firefly inspired controller placement algorithm in software defined network,” *In 2019 IEEE 2nd international conference on computer and communication engineering technology (CCET)*, 2019. pp. 254-

258. https://doi.org/10.1109/CCET48361.2019.8989320.

[7] M. Dorigo, M. Birattari and T. Stutzle, "Ant colony optimization artificial ants as a computational intelligence technique," *IEEE Comput Intell M*, vol. 1, no. 4, pp. 28–39, 2006. https://doi.org/10.1109/MCI.2006.329691.

[8] D. Karaboga and B. Basturk, "Artificial bee colony (ABC) optimization algorithm for solving constrained optimization," *Foundations of fuzzy logic and soft computing, Lecture Notes in Computer Science*, vol. 4529, 2007, pp.789–798. https://doi.org/10.1007/978-3-540-72950-1_77.

[9] M. A. Ahandani, R. Banimahd and N. P. Shrjoposht, "Solving the parameter identification problem using shuffled frog leaping with opposition-based initialization," In 2011 1st International eConference on Computer and Knowledge Engineering (ICCKE), 2011, pp. 49-53. https://doi.org/10.1109/ICCKE.2011.6413323.

[10] B. Xing, W. J. Gao, "Imperialist competitive algorithm," in Innovative computational intelligence: A rough guide to 134 clever algorithms, pp. 203-209 . 2014.

[11] A. Kalananda, V. K.Reddy, and V. L. Narayana Komanapalli, "A competitive learning-based Grey wolf Optimizer for engineering problems and its application to multi-layer perceptron training," *Multimedia Tools and Applications*, vol. 82, pp. 1-59, 2023. https://doi.org/10.1007/s11042-023-15146-x.

[12] W. T. Lunardi, H. Voos and L. H. Cherri, "An effective hybrid imperialist competitive algorithm and tabu search for an extended flexible job shop scheduling problem," In Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing, 2019, pp. 204-211. https://doi.org/10.1145/3297280.3297302.

[13] R. Rajabioun, E. Atashpaz-Gargari and C. Lucas, "Colonial competitive algorithm as a tool for Nash equilibrium point achievement," In Computational Science and Its Applications–ICCSA 2008: International Conference, 2008, pp. 680-695. https://doi.org/10.1007/978-3-540-69848-7_55.

[14] M. Eusuff, K. Lansey and F. Pasha, "Shuffled frog-leaping algorithm: a memetic meta-heuristic for discrete optimization," *Engineering optimization*, vol. 38, no. 2, pp. 129-154, 2006. https://doi.org/10.1080/03052150500384759.

[15] B. B. Maaroof, T. A. Rashid, J. M. Abdulla, B. A. Hassan, A. Alsadoon, M. Mohammadi, M. Khishe and S. Mirjalili, "Current studies and applications of shuffled frog leaping algorithm: a review," *Archives of Computational Methods in Engineering*, vol. 29, no. 5, pp. 3459-3474, 2022. https://doi.org/10.1007/s11831-021-09707-2.

[16] M. Eusuff, K. Lansey, and F. Pasha, "Shuffled frog-leaping algorithm: a memetic meta-heuristic for discrete optimization," *Engineering optimization*, vol. 38, no. 2, pp. 129-154, 2006. https://doi.org/10.1080/03052150500384759.

[17] X. Yang, X. He, "Firefly algorithm: recent advances and applications," *International journal of swarm intelligence*, vol. 1, no. 1, pp. 36-50, 2013. https://doi.org/10.1504/IJSI.2013.055801.

[18] N. F. Johari, et al. and A. Udin, "Firefly algorithm for optimization problem," *Applied Mechanics and Materials*, vol. 421, pp. 512-517, 2013. https://doi.org/10.4028/www.scientific.net/AMM.421.512.

[19] X. Yang, "Firefly algorithm, stochastic test functions and design optimisation," *International journal of bio-inspired computation*, vol. 2, no. 2, pp. 78-84, 2010. https://doi.org/10.1504/IJBIC.2010.032124.

[20] S. Farahani, A. A. Abshouri, B. Nasiri, and M. R. Meybodi, "A Gaussian firefly algorithm," *International Journal of Machine Learning and Computing*, vol. 1, no. 5, p. 448, 2011.

[21] V. Kumar, D. Kumar, "A systematic review on firefly algorithm: past, present, and future," Archives of Computational Methods in Engineering, vol. 28, pp. 3269-3291, 2021. https://doi.org/10.1007/s11831-020-09498-y.

[22] X. Yang, "Firefly algorithms for multimodal optimization," *International symposium on stochastic algorithms*, Springer, 2009, pp. 169-178. https://doi.org/10.1007/978-3-642-04944-6_14.

[23] M. H. Alizadeh, A. Toloei, "Designing Pitch Angle Compensator for an UAV and Robustification it with Bee Colony Optimization Algorithm," *Technology in Aerospace Engineering*, pp. 1-13, 2023. https://doi.org/10.22034/jtae.2024.8.1.1.

[24] B. Attaran, M. Zarchi and A. R. Toloei, "Numerical survey of vibrational model for third aircraft based on HR suspension system actuator using two bee algorithm objective functions," *International Journal of Engineering*, vol. 30, no. 6, pp. 887-894, 2017.

[25] A. R. Toloei, M. Zarchi and B. Attaran, "Application of active suspension system to reduce aircraft vibration using pid technique and bees algorithm," *International Journal of Computer Applications*, vol. 98, no. 6, 2014. 10.5120/17187-7368.

[26] A. R. Toloei, M. Zarchi and B. Attaran, "Vibration control of aircraft semi-active suspension system using PID-bees technique," *International Journal of Computer Applications*, vol. 99, no. 10, 2014. https://doi.org/10.5120/17408-7980.

[27] M. Ameri and E. Behrouzian Nejad, "Presenting a clustering method based on colonial competition algorithm to increase network lifetime in wireless sensor networks," *In 2nd international conference on Research in Engineering*, Science and Technology, 2005. (InPersian)

[28] Z. Ghafari Moghadam, M. Hashemi Tabar, and E. Moradi, "Water DemandManagement Using Dynamic Firefly Optimization Algorithm: A Case Study of Hirmand River," *Scientific Quarterly Journal of Water Resources Engineering*, vol. 13, no. 46, pp. 85-98, 2020. (InPersian)

[29] H. Zhu, Y. Hu and W. Zhu, "A dynamic adaptive particle swarm optimization and genetic algorithm for different constrained engineering design optimization problems," *Advances in Mechanical Engineering*, vol. 11, no. 3, 2019. https://doi.org/10.1177/1687814018824930.

[30] C. I. Okonta, et al. and E. D. Okelue, "A heuristic based ant colony optimization algorithm for energy efficient smart homes," *In Proc. 5th International Conference Exhib. Clean Energy*, pp. 1-12, 2016.